Paralelné algoritmy, časť č. 11

František Mráz

Kabinet software a výuky informatiky, MFF UK, Praha

Paralelné algoritmy, 2011/2012

Obsah

Outline

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \ldots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \ldots, n$

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \ldots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \ldots, n$
- Použijeme 2 typy položiek:

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \ldots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \ldots, n$
- Použijeme 2 typy položiek:
  1. (A, i, j), kde $A \in \Pi$, $0 \leq i < j \leq n$, (“sukňa”)

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \ldots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \ldots, n$
- Použijeme 2 typy položiek:
  1. (A, i, j), kde $A \in \Pi$, $0 \leq i < j \leq n$, (“sukňa”)
  2. (A, i, j, B, k, l), kde $A, B \in \Pi$, $0 \leq i \leq k < l \leq j \leq n$, (“nohavice”)

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \ldots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \ldots, n$
- Použijeme 2 typy položiek:
  1. (A, i, j), kde $A \in \Pi$, $0 \leq i < j \leq n$, (“sukňa”)
  2. (A, i, j, B, k, l), kde $A, B \in \Pi$, $0 \leq i \leq k < l \leq j \leq n$, (“nohavice”)
- Položka p je *platná*, ak

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \ldots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \ldots, n$
- Použijeme 2 typy položiek:
  1. (A, i, j), kde $A \in \Pi$, $0 \leq i < j \leq n$, (“sukňa”)
  2. (A, i, j, B, k, l), kde $A, B \in \Pi$, $0 \leq i \leq k < l \leq j \leq n$, (“nohavice”)
- Položka p je *platná*, ak
  1. $p = (A, i, j)$, a $A \Rightarrow_G^* a_{i+1} \ldots a_j$

Rozpoznávanie bezkontexového jazyka

Nech $G = (\Pi, \Sigma.S, P)$ je daná bezkontextová gramatika.

Zadanie: slovo w z Σ^*.

Otázka: $w \stackrel{?}{\in} L(G)$.

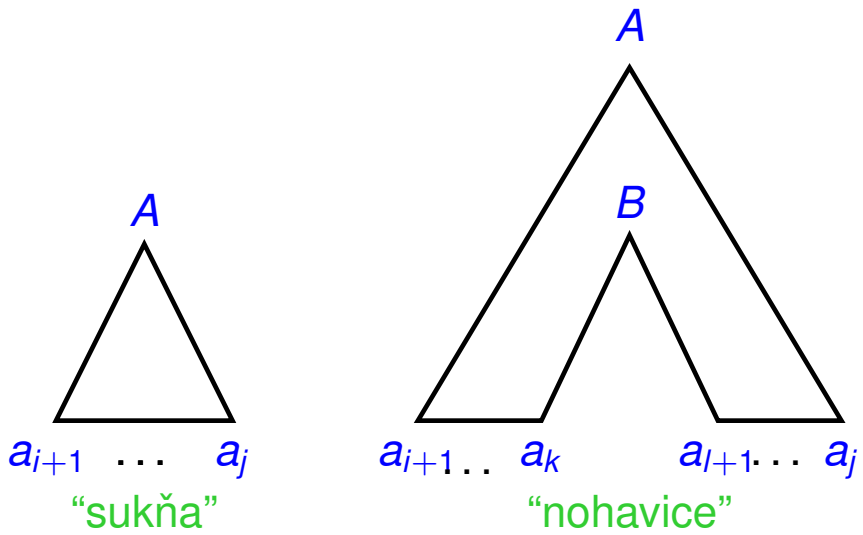
Veta

Ľubovoľný bezkontextový jazyk sa dá rozpoznávať v čase $O(\log n)$ s $O(n^6)$ procesormi, kde n je dĺžka vstupného slova.

Dôkaz:

- Nech $w = a_1 a_2 \dots a_n$, kde $a_i \in \Sigma$, pre $i = 1, \dots, n$
- Použijeme 2 typy položiek:
  1. (A, i, j), kde $A \in \Pi$, $0 \leq i < j \leq n$, (“sukňa”)
  2. (A, i, j, B, k, l), kde $A, B \in \Pi$, $0 \leq i \leq k < l \leq j \leq n$, (“nohavice”)
- Položka p je *platná*, ak
  1. $p = (A, i, j)$, a $A \Rightarrow_G^* a_{i+1} \dots a_j$
  2. $p = (A, i, j, B, k, l)$ a $A \Rightarrow_G^* a_{i+1} \dots a_k B a_{l+1} \dots a_j$

Rozpoznávanie bezkontexového jazyka



veľkosť položky = počet terminálov

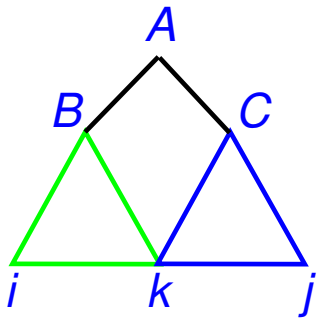
1. ak $p = (A, i, j)$, potom $size(p) = j - i$
2. ak $p = (A, i, j, B, k, l)$, potom $size(p) = k - i + j - l$

- budeme predpokladať, že G je Chomského normálnej forme – iba pravidlá tvaru $X \to YZ, X \to a$, kde $X, Y, Z \in \Pi, a \in \Sigma$

Skladanie položiek zo sukní

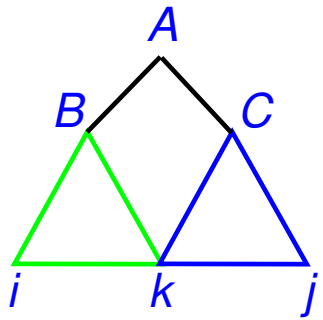
- Ak $A \to BC \in P$, $x = (A, i, j)$, $y = (B, i, k)$, $z = (C, k, j)$, potom
  ak y, z sú platné, tak x je platné

  budeme to označovať $y, z \vdash x$ a $z, y \vdash x$



Skladanie položiek zo sukní

- Ak $A \to BC \in P$, $x = (A, i, j)$, $y = (B, i, k)$, $z = (C, k, j)$, potom
  ak y, z sú platné, tak x je platné
  budeme to označovať $y, z \vdash x$ a $z, y \vdash x$



- **Algoritmus** (M je množina položiek – sukní):
  $M := \{(A, i, i+1) \mid A \to a_{i+1} \in P\}$
  repeat "several" **times**
    for all položky x, y, z in parallel **do**
      if $y, z \in M$ **and** $y, z \vdash x$ **then** $M := M \cup \{x\}$
  if $(S, 0, n) \in M$ **then** accept

Skladanie sukní je pomalé

- **Algoritmus** (M je množina položiek – sukní):
  $M := \{(A, i, i+1) \mid A \to a_{i+1} \in P\}$
  repeat "several" **times**
    for all položky x, y, z in parallel **do**
      if $y, z \in M$ **and** $y, z \vdash x$ **then** $M := M \cup \{x\}$
  if $(S, 0, n) \in M$ **then** accept

Skladanie sukní je pomalé

- **Algoritmus** (M je množina položiek – sukní):
  $M := \{(A, i, i+1) \mid A \to a_{i+1} \in P\}$
  repeat "several" **times**
    for all položky x, y, z in parallel **do**
      if $y, z \in M$ **and** $y, z \vdash x$ **then** $M := M \cup \{x\}$
  if $(S, 0, n) \in M$ **then** accept
- pre gramatiku s pravidlami
  $S \to AS$
  $S \to a$
  $A \to a$
  "several" musí byť $\Omega(n)$

Skladanie sukní je pomalé

- **Algoritmus** (M je množina položiek – sukní):
  $M := \{(A, i, i+1) \mid A \to a_{i+1} \in P\}$
  repeat "several" **times**
    for all položky x, y, z in parallel **do**
      if $y, z \in M$ **and** $y, z \vdash x$ **then** $M := M \cup \{x\}$
  if $(S, 0, n) \in M$ **then** accept
- pre gramatiku s pravidlami
  $S \to AS$
  $S \to a$
  $A \to a$
  "several" musí byť $\Omega(n)$
- $\Rightarrow$ aby sme rozpoznávanie zrýchlili, musíme skladať aj zložené položky – nohavice

Outline

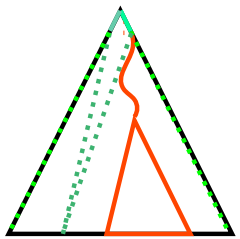
Rozpoznávanie bezkontexového jazyka

Lemma

Ak p je platná položka veľkosti $k > 1$, p sa dá "zložiť" z konečného počtu (max. 4) položiek veľkosti maximálne $\frac{2}{3}k$.

Dôkaz:

(i) ak $p = (A, i, j)$, potom v derivačnom strome pre p existuje podstrom (sukňa) veľkosti medzi $\frac{1}{3}k$ a $\frac{2}{3}k$ a p sa dá zložiť z tohoto podstromu a príslušných nohavíc



Rozpoznávanie bezkontexového jazyka

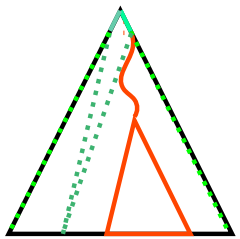
Lemma

Ak p je platná položka veľkosti $k > 1$, p sa dá "zložiť" z konečného počtu (max. 4) položiek veľkosti maximálne $\frac{2}{3}k$.

Dôkaz:

(i) ak $p = (A, i, j)$, potom v derivačnom strome pre p existuje podstrom (sukňa) veľkosti medzi $\frac{1}{3}k$ a $\frac{2}{3}k$ a p sa dá zložiť z tohoto podstromu a príslušných nohavíc

- koreň p má presne 2 synov (Chomského normálna forma)



Rozpoznávanie bezkontexového jazyka

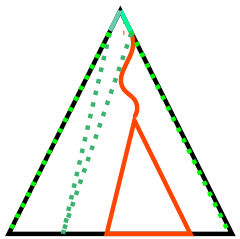
Lemma

Ak p je platná položka veľkosti $k > 1$, p sa dá "zložiť" z konečného počtu (max. 4) položiek veľkosti maximálne $\frac{2}{3}k$.

Dôkaz:

(i) ak $p = (A, i, j)$, potom v derivačnom strome pre p existuje podstrom (sukňa) veľkosti medzi $\frac{1}{3}k$ a $\frac{2}{3}k$ a p sa dá zložiť z tohoto podstromu a príslušných nohavíc

- koreň p má presne 2 synov (Chomského normálna forma)
- z koreňa A pôjdeme do väčšieho z dvoch podstromov (prípadne do ľubovoľného, ak majú obidva podstromy rovnakú veľkosť)



Rozpoznávanie bezkontexového jazyka

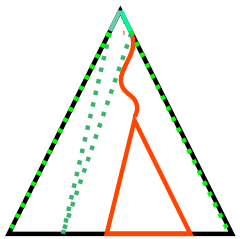
Lemma

Ak p je platná položka veľkosti $k > 1$, p sa dá "zložiť" z konečného počtu (max. 4) položiek veľkosti maximálne $\frac{2}{3}k$.

Dôkaz:

(i) ak $p = (A, i, j)$, potom v derivačnom strome pre p existuje podstrom (sukňa) veľkosti medzi $\frac{1}{3}k$ a $\frac{2}{3}k$ a p sa dá zložiť z tohoto podstromu a príslušných nohavíc

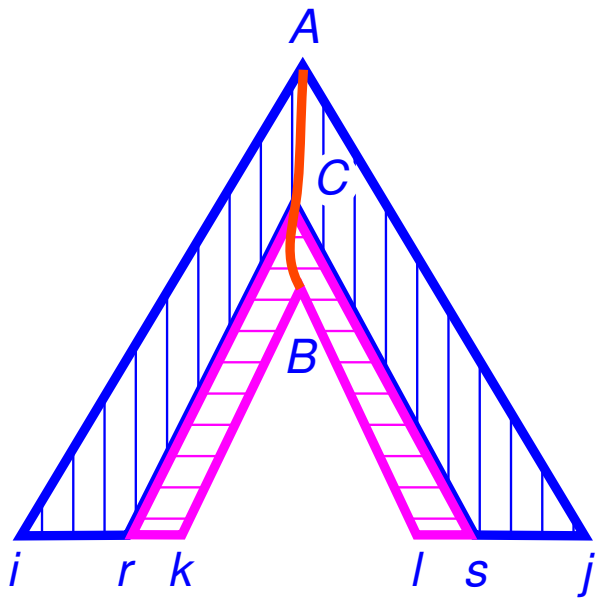
- koreň p má presne 2 synov (Chomského normálna forma)
- z koreňa A pôjdeme do väčšieho z dvoch podstromov (prípadne do ľubovoľného, ak majú obidva podstromy rovnakú veľkosť)
- ak tento podstrom má veľkosť $> \frac{2}{3}k$, tak pokračujeme rekurzívne hlbšie



Rozpoznávanie bezkontexového jazyka

Dôkaz (pokračovanie):

(ii) ak $p = (A, i, j, B, k, l)$, potom v derivačnom strome pôjdeme od koreňa po ceste k "diere" (tj. koreňu sukne (B, k, l)).

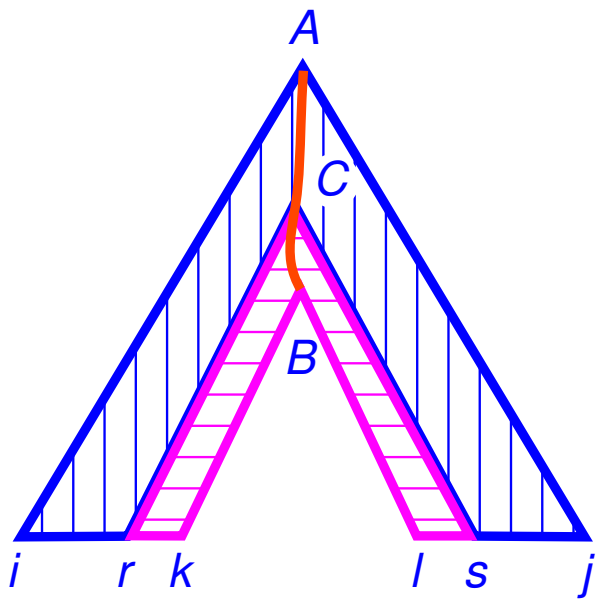


Rozpoznávanie bezkontexového jazyka

Dôkaz (pokračovanie):

(ii) ak $p = (A, i, j, B, k, l)$, potom v derivačnom strome pôjdeme od koreňa po ceste k "diere" (tj. koreňu sukne (B, k, l)).

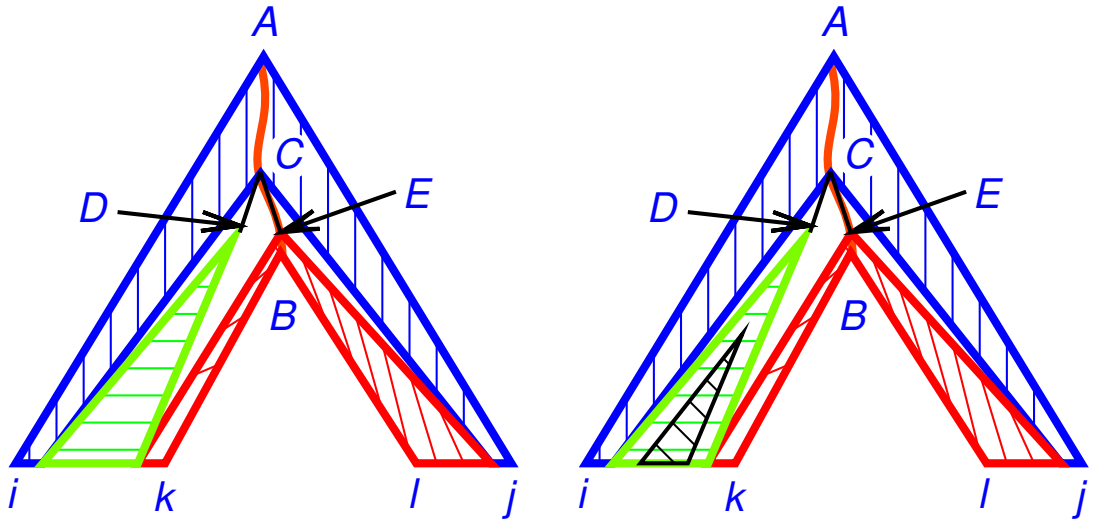
a) Ak na tejto ceste leží uzol (C, r, s), pod ktorým sú nohavice veľkosti medzi $\frac{1}{3}k$ a $\frac{2}{3}k$, tak sa p dá zložiť z dvoch nohavíc (A, i, j, C, r, s) a (C, r, s, B, k, l) veľkosti medzi $\frac{1}{3}k$ a $\frac{2}{3}k$.



Rozpoznávanie bezkontexového jazyka

Dôkaz (pokračovanie):

(ii) ak $p = (A, i, j, B, k, l)$, potom v derivačnom strome pôjdeme od koreňa po ceste k "diere" (tj. koreňu sukne (B, k, l)).

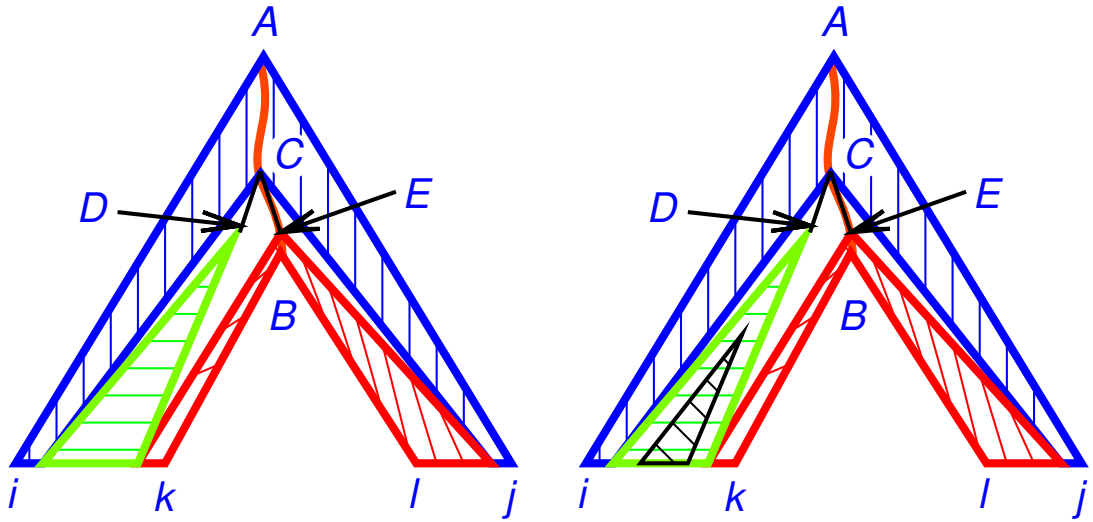


Rozpoznávanie bezkontexového jazyka

Dôkaz (pokračovanie):

(ii) ak $p = (A, i, j, B, k, l)$, potom v derivačnom strome pôjdeme od koreňa po ceste k "diere" (tj. koreňu sukne (B, k, l)).

b) Inak existuje na tejto ceste uzol zodpovedajúci položke (C, r, s) taký, že nohavice $size((A, i, j, C, r, s)) < \frac{1}{3}k$ a syn E uzlu C, ktorý leží na tejto ceste, zodpovedá nohaviciam s veľkosťou menšou než $\frac{1}{3}k$. Potom sukňa, ktorá je pod druhým synom uzlu C, má veľkosť



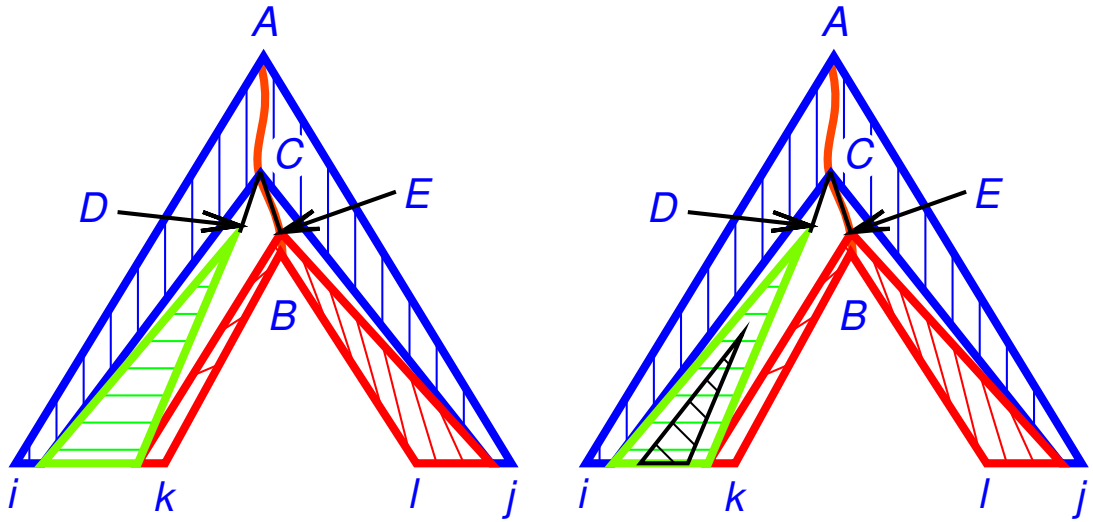
Rozpoznávanie bezkontexového jazyka

Dôkaz (pokračovanie):

(ii) ak $p = (A, i, j, B, k, l)$, potom v derivačnom strome pôjdeme od koreňa po ceste k "diere" (tj. koreňu sukne (B, k, l)).

b) Inak existuje na tejto ceste uzol zodpovedajúci položke (C, r, s) taký, že nohavice $size((A, i, j, C, r, s)) < \frac{1}{3}k$ a syn E uzlu C, ktorý leží na tejto ceste, zodpovedá nohaviciam s veľkosťou menšou než $\frac{1}{3}k$. Potom sukňa, ktorá je pod druhým synom uzlu C, má veľkosť

- buď $\leq \frac{2}{3}k$ – hotovo,



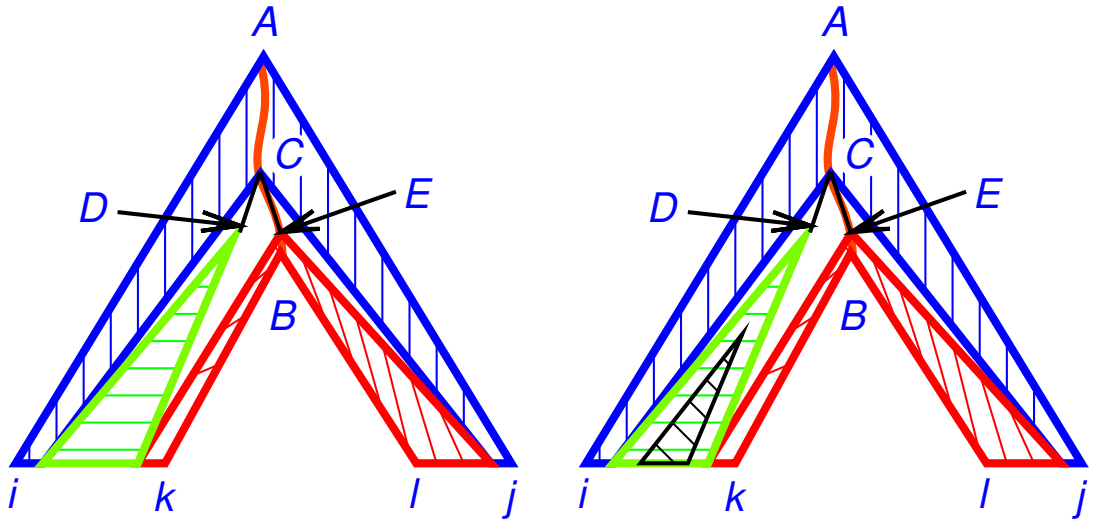
Rozpoznávanie bezkontexového jazyka

Dôkaz (pokračovanie):

(ii) ak $p = (A, i, j, B, k, l)$, potom v derivačnom strome pôjdeme od koreňa po ceste k "diere" (tj. koreňu sukne (B, k, l)).

b) Inak existuje na tejto ceste uzol zodpovedajúci položke (C, r, s) taký, že nohavice $size((A, i, j, C, r, s)) < \frac{1}{3}k$ a syn E uzlu C, ktorý leží na tejto ceste, zodpovedá nohaviciam s veľkosťou menšou než $\frac{1}{3}k$. Potom sukňa, ktorá je pod druhým synom uzlu C, má veľkosť

- buď $\leq \frac{2}{3}k$ – hotovo,
- alebo $> \frac{2}{3}k$ – potom sa dá zložiť z nohavíc a sukne veľkosti maximálne $\frac{2}{3}k$ (viz. (i))



Rýchly algoritmus [Rytter, 1986]

pre $x = (A, i, j)$, $y = (B, k, l)$ bude $< x, y >$ označovať nohavice (A, i, j, B, k, l)

$M := \emptyset$
for all i, $0 \leq i < n, A \in \Pi$ také, že $A \to a_{i+1} \in P$
in parallel do vlož $(A, i, i+1)$ do M
repeat "several" **times**
{aktivácia}
for all x, y, z, také že $y, z \vdash x$ a $z \in M$
in parallel do vlož $< x, y >$ do M
{zdvojenie nohavíc}
for all x, y, z, také že $< x, z > \in M < z, y > \in M$
in parallel do vlož $< x, y >$ do M
{zdvojenie nohavíc}
for all x, y, z, také že $< x, z > \in M < z, y > \in M$
in parallel do vlož $< x, y >$ do M
{zloženie nohavíc a sukne}
for all x, y, také že $< x, z > \in M$ a $z \in M$
in parallel do vlož $< x >$ do M
if $(S, 0, n) \in M$ **then** accept

- "zdvojenie nohavíc" zmenší počet iterácií – viz časť (ii) b) predchádzajúceho dôkazu

Rýchly algoritmus [Rytter, 1986]

pre $x = (A, i, j)$, $y = (B, k, l)$ bude $< x, y >$ označovať nohavice (A, i, j, B, k, l)

$M := \emptyset$
for all i, $0 \leq i < n, A \in \Pi$ také, že $A \to a_{i+1} \in P$
in parallel do vlož $(A, i, i+1)$ do M
repeat "several" **times**
{aktivácia}
for all x, y, z, také že $y, z \vdash x$ a $z \in M$
in parallel do vlož $< x, y >$ do M
{zdvojenie nohavíc}
for all x, y, z, také že $< x, z > \in M < z, y > \in M$
in parallel do vlož $< x, y >$ do M
{zdvojenie nohavíc}
for all x, y, z, také že $< x, z > \in M < z, y > \in M$
in parallel do vlož $< x, y >$ do M
{zloženie nohavíc a sukne}
for all x, y, také že $< x, z > \in M$ a $z \in M$
in parallel do vlož $< x >$ do M
if $(S, 0, n) \in M$ **then** accept

- "zdvojenie nohavíc" zmenší počet iterácií – viz časť (ii) b) predchádzajúceho dôkazu
- "several" stačí $O(\log n)$, dokonca sa dá ukázať, že stačí $\log n$

Rýchly algoritmus [Rytter, 1986]

pre $x = (A, i, j)$, $y = (B, k, l)$ bude $< x, y >$ označovať nohavice (A, i, j, B, k, l)

$M := \emptyset$
for all i, $0 \leq i < n, A \in \Pi$ také, že $A \to a_{i+1} \in P$
in parallel do vlož $(A, i, i+1)$ do M
repeat "several" **times**
{aktivácia}
for all x, y, z, také že $y, z \vdash x$ a $z \in M$
in parallel do vlož $< x, y >$ do M
{zdvojenie nohavíc}
for all x, y, z, také že $< x, z > \in M < z, y > \in M$
in parallel do vlož $< x, y >$ do M
{zdvojenie nohavíc}
for all x, y, z, také že $< x, z > \in M < z, y > \in M$
in parallel do vlož $< x, y >$ do M
{zloženie nohavíc a sukne}
for all x, y, také že $< x, z > \in M$ a $z \in M$
in parallel do vlož $< x >$ do M
if $(S, 0, n) \in M$ **then** accept

- "zdvojenie nohavíc" zmenší počet iterácií – viz časť (ii) b) predchádzajúceho dôkazu
- "several" stačí $O(\log n)$, dokonca sa dá ukázať, že stačí $\log n$
- Celkovo: $O(n^6)$ procesorov a čas $O(\log n)$

Outline

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

$A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A, i, j)$ je platná položka (sukňa)

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A,i,j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A, i, j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A, i, j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A, i, j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný
  2. $Left[x]$ – ľavý syn vrcholu x,

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A,i,j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný
  2. $Left[x]$ – ľavý syn vrcholu x,
  3. $Right[x]$ – pravý syn vrcholu x,

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A, i, j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný
  2. $Left[x]$ – ľavý syn vrcholu x,
  3. $Right[x]$ – pravý syn vrcholu x,
- ďalej potrebujeme určiť $Label[x]$ taký, že:

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku
  $$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$
  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A,i,j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný
  2. $Left[x]$ – ľavý syn vrcholu x,
  3. $Right[x]$ – pravý syn vrcholu x,
- ďalej potrebujeme určiť $Label[x]$ taký, že:
  1. $Label[root] = S$, kde S je počiatočný neterminál G,

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A,i,j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný
  2. $Left[x]$ – ľavý syn vrcholu x,
  3. $Right[x]$ – pravý syn vrcholu x,
- ďalej potrebujeme určiť $Label[x]$ taký, že:
  1. $Label[root] = S$, kde S je počiatočný neterminál G,
  2. $Label[x] \to Label[Left[x]]Label[Right[x]]$ je pravidlo G,

Analýza bezkontextového jazyka

- nestačí vedieť či $w \in L(G)$, ale chceme aj derivačný strom odvodenia w podľa G
- môžeme predpokladať, že po rozpoznaní slova $w = a_1 \dots a_n$ máme tabuľku

$$Tab[i,j] = \begin{cases} \{A \mid A \Rightarrow^* a_{i+1} \dots a_j\} & \text{ak } i < j \\ \emptyset & \text{inak} \end{cases}$$

  $A \Rightarrow^* a_{i+1} \cdots a_j \Leftrightarrow (A,i,j)$ je platná položka (sukňa)
- na určenie derivačného stromu stačí spočítať orientovaný binárny strom PT s $2n-1$ uzlami taký, že každý vrchol okrem koreňa má vstupný stupeň 1, koreň $root$ má vstupný stupeň 0
- pre každý (vnútorný) uzol x stromu PT potrebujeme
  1. $Father[x]$ – otec vrcholu x, $Father[root]$ je nedefinovaný
  2. $Left[x]$ – ľavý syn vrcholu x,
  3. $Right[x]$ – pravý syn vrcholu x,
- ďalej potrebujeme určiť $Label[x]$ taký, že:
  1. $Label[root] = S$, kde S je počiatočný neterminál G,
  2. $Label[x] \to Label[Left[x]]Label[Right[x]]$ je pravidlo G,
  3. $Label[x] = a_i$, ak x je i-ty list stromu PT

Analýza bezkontextového jazyka

- nech Q je vektor boooleovských hodnôt dĺžky n taký, že aspoň jeden jeho prvok je true
  $first(Q) = \min\{k \mid 0 \leq k \leq n, Q[k] = true\}$
  $first(Q)$ sa dá spočítať v čase $O(\log n)$ s n procesormi

Analýza bezkontextového jazyka

- nech Q je vektor boooleovských hodnôt dĺžky n taký, že aspoň jeden jeho prvok je true
  $first(Q) = \min\{k \mid 0 \leq k \leq n, Q[k] = true\}$
  $first(Q)$ sa dá spočítať v čase $O(\log n)$ s n procesormi
- nech $A \in \Pi$, $X, Y \subseteq \Pi$
  $find(A, X, Y) = (B, C)$, ak (B, C) je lexikograficky najmenšia dvojica neterminálov taká, že $B \in X$, $C \in Y$ a $A \to BC \in P$, ak také (B, C) neexistuje, tak $find(A, X, Y)$ vráti *undefined*
  dá sa spočítať v čase $O(1)$ s jediným procesorom

Analýza bezkontextového jazyka

- nech Q je vektor boooleovských hodnôt dĺžky n taký, že aspoň jeden jeho prvok je true
  $first(Q) = \min\{k \mid 0 \leq k \leq n, Q[k] = true\}$
  $first(Q)$ sa dá spočítať v čase $O(\log n)$ s n procesormi
- nech $A \in \Pi$, $X, Y \subseteq \Pi$
  $find(A, X, Y) = (B, C)$, ak (B, C) je lexikograficky najmenšia dvojica neterminálov taká, že $B \in X$, $C \in Y$ a $A \to BC \in P$, ak také (B, C) neexistuje, tak $find(A, X, Y)$ vráti $undefined$
  dá sa spočítať v čase $O(1)$ s jediným procesorom
- spočítame vektory $mark_{ij}$;
  $mark_{ij}[A, k] = true \equiv 0 \leq i < k < j \leq n$ a
  $find(A, Tab[i, k], Tab[k, j]) \neq undefined$

Analýza bezkontextového jazyka

- pre každú sukňu výšky > 1 vyberieme ľavého a pravého syna – pomocou *mark* a *find*

Veta

Nech je možné rozpoznávať ľubovoľný bezkontextový jazyk v čase $T(n)$ s $R(n)$ procesormi a $T(n) = \Omega(\log n)$. Potom derivačný strom odvodenia vstupného slova je možné spočítať na COMMON PRAM v čase $O(T(n))$ s $O(R(n)n^2 + n^3)$ procesormi. Ak rozpoznávací algoritmus skonštruuje CYK tabuľku (Tab), tak stačí $O(R(n) + n^3)$ procesorov.

Analýza bezkontextového jazyka

- pre každú sukňu výšky > 1 vyberieme ľavého a pravého syna – pomocou *mark* a *find*
- do *PT* dáme *root* a všetky uzly dosiahnuteľné z *root* cez ľavých a pravých synov

Veta

Nech je možné rozpoznávať ľubovoľný bezkontextový jazyk v čase $T(n)$ s $R(n)$ procesormi a $T(n) = \Omega(\log n)$. Potom derivačný strom odvodenia vstupného slova je možné spočítať na COMMON PRAM v čase $O(T(n))$ s $O(R(n)n^2 + n^3)$ procesormi. Ak rozpoznávací algoritmus skonštruuje CYK tabuľku (Tab), tak stačí $O(R(n) + n^3)$ procesorov.